

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
ALGORITMOS E ESTRUTURAS DE DADOS III

2º Período Emergencial de 2020

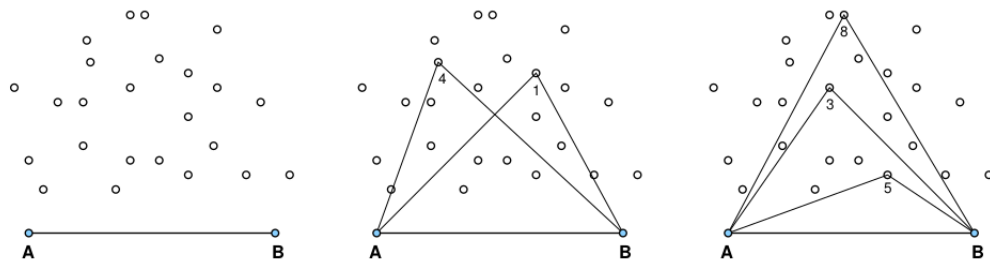
Professor: Leonardo Chaves Dutra da Rocha

Trabalho Prático 1

Data de Entrega: 05 de Março 2020.

Hipercampos

São dadas duas âncoras, dois pontos $A = (X_A, 0)$ e $B = (X_B, 0)$, formando um segmento horizontal, tal que $0 < X_A < X_B$, e um conjunto P de N pontos da forma (X, Y) , tal que $X > 0$ e $Y > 0$. A figura mais à esquerda exemplifica uma possível entrada.



Para "ligar" um ponto $v \in P$ precisamos desenhar os dois segmentos de reta (v, A) e (v, B) . Queremos ligar vários pontos, mas de modo que os segmentos se interceptem apenas nas âncoras. Por exemplo, a figura do meio mostra dois pontos, 1 e 4, que não podem estar ligados ao mesmo tempo, pois haveria interseção dos segmentos fora das âncoras. A figura mais à direita mostra que é possível ligar pelo menos 3 pontos, 8, 5 e 3, com interseção apenas nas âncoras. Seu programa deve computar o número máximo de pontos que é possível ligar com interseção de segmentos apenas nas âncoras.

Entrada

A primeira linha da entrada contém três inteiros, N ($1 \leq N \leq 100$), X_A e X_B ($0 < X_A < X_B \leq 10^4$) representando, respectivamente, o número de pontos no conjunto P e as abscissas das âncoras A e B . As N linhas seguintes contêm, cada uma, dois inteiros X_i e Y_i ($0 < X_i, Y_i \leq 10^4$), representando as coordenadas dos pontos, para $1 \leq i \leq N$. Não há pontos coincidentes e não há dois pontos u e v distintos tais que A, u, v ou B, u, v sejam colineares.

Exemplos de entrada:

```
4 1 10
2 4
5 1
6 5
7 8
```

```
2 2 8
3 4
7 4
```

Saída

Seu programa deve imprimir uma linha contendo um inteiro, representando o número máximo de pontos de P que podem ser ligados com interseção de segmentos apenas nas âncoras.

Exemplos de saída

3
1

Elabore um programa com as seguintes características:

1. O programa deve ser implementado em módulos. Deve ainda ser definido um arquivo de protótipos e definições contendo as estruturas de dados e protótipos das funções dos vários módulos. A compilação do programa deve utilizar o utilitário Make.
2. Todas as estruturas de dados devem ser alocadas dinamicamente, assim como devem ser desalocadas após o processamento. As rotinas de teste de colisão também devem ser implementadas usando estruturas de alocação dinâmica.
3. O programa deve receber dois parâmetros pela linha de comando, utilizando a primitiva `getopt`:
 - (a) -i arquivo de entrada
 - (b) -o arquivo de saída

O programa implementado deve ser avaliado para várias arquivos de entrada utilizando as funções `getrusage` e `gettimeofday`. Deve-se também distinguir entre os tempos de computação e tempos de entrada e saída. Comente sobre os tempos de usuário e os tempos de sistema e sua relação com os tempos de relógio.

Avaliação

Deverão ser entregues:

- listagem das rotinas;
- descrição breve dos algoritmos e das estruturas de dados utilizadas;
- análise da complexidade das rotinas;
- análise dos resultados obtidos.

Distribuição dos pontos:

- execução (E)
execução correta: 80%
- estilo de programação
código bem estruturado: 10%
código legível: 10%
- documentação (D)
comentários explicativos: 40%
análise de complexidade: 30%
análise de resultados: 30%

A nota final é calculada como a média harmônica entre execução (E) e documentação (D):

$$\frac{D * E}{\frac{D+E}{2}}$$